Service Components

more...

The neighborhood service is a distributed system. Neighborhood servers are the core components of the system. Clients are the user interfaces, and auxiliary components gather information about users and the Web.

Servers

Neighborhood servers provide the user dimension of the virtual world. They serve neighborhood clients in the same sense as Web servers serve Web clients. The Web is segmented in parts, which are created by individual Web servers. A neighborhood server is responsible for the user space of a part of the Web and it will create the user space for this part. In this case each Webserver is accompanied or complemented by a neighborhood server. But a neighborhood server can also create the user space for a group of Web servers.

A neighborhood server essentially maintains 2 databases, a link datbase and a user database. The link database reflects the document and link structure of the augmented Web server(s). Neighborhood servers maintain backward references for all hypertext references in order to provide symmetric visibility. Links to objects on the same server (core links) are maintained easily. Links to documents on other Web servers (surface links) generate network traffic between neighborhood servers. The LINK-command (ASSOCIATE locations) is used to announce a link. This is done very rarely so that the generated traffic is only a small fraction of the HTTP traffic between Web clients and Web servers.

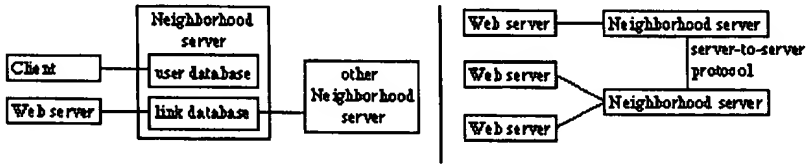

Figure 5: Left: a simplified view of a neighborhood server. It communicates with clients, Web servers, and other neighborhood servers. Right: A neighborhood server attends one or more Web servers.

The main task of neighborhood servers is the computation of the visible objects. This is done based on the Web link structure, user preferences, and various system settings. User preferences and site specific settings control the computation of the visibility-function and as such the set of visible objects. We are currently using a visibility-function box-shaped in space and time. The person's instance variable "link-distance" is the length of the box in space. This parameter is typically set to 2, which means that other persons, which are less or equal than 2 hypertext references apart from the user's position are visible (see figure 6).

The second parameter controls the depth of the visibility-function in time. The time dimension has not been shown in figure 4, but actually proved very helpful to increase the usability of the system. The time dimension of the visibility-function smoothes the movement between pages. It is much easier for the users to observe the movement if persons remain registered with locations for a short time after they left a Web page.

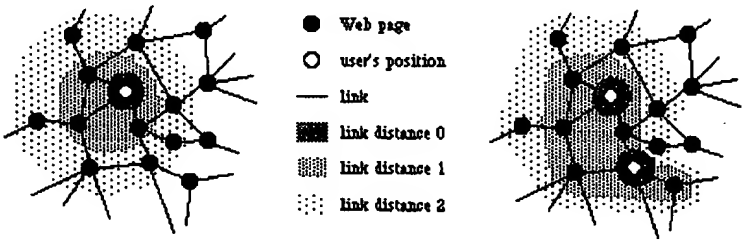

Figure 6: Pages in the visible area depending on the link-distance

variable. Right: the shape of the visibility-function can be complex, if a user is browsing multiple pages simultaneously.

The result of the computation, a list of visible pages, persons, and communications is kept available for neighborhood clients.

Clients

Neighborhood clients presents a dynamically changing view of the user space. They are using the protocol introduced above to retrieve information about the position of the user, the surroundings, other users and their properties.

Basic User Interface

The basic user interface originates from a test tool for the protocol engine and the neighborhood server. However, it is very useful as a lean user interface. It shows the neighbors as a list with icons, nicknames, and the position of the users (as URLs). The user interface allows modifications to instance variables by the user who owns the person-object in the neighborhood server.

The basic user interface is implemented in Java. The code is separated into two distinct layers. The lower layer (network layer) is a protocol handler, which implements the neighborhood protocol. We are currently using an encapsulation of the protocol in HTTP in order to tunnel through proxies and firewalls. The upper layer (display layer) consists of display classes. The display classes show the content of instance variables of specific objects of the neighborhood server, i.e names, icons and positions of the neighbors.

The display layer is independent of the protocol implementation. It will survive a major redesign of the protocol as long as the basic concepts remain the same.

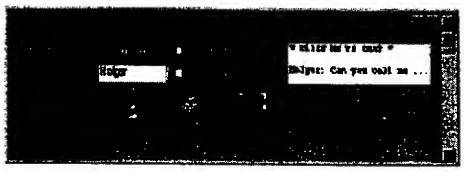

Figure 7: The basic user interface. Neighbors are shown as icons. The icons are click-able to initiate communication. The little chat-window in the upper right tries to attract people to ongoing communication.

Advanced User Interface

The VRML 2.0 standard and an external authoring interface (EAI) enable dynamic virtual reality displays. Rather than being just a list of visible persons the neighborhood can be presented as a three dimensional visualization of a (small) fraction of the Web. This is similar to [Domel94], "A Graphical Hypertext Navigation Tool", but employs a VRML engine for the rendering of 3-dimensional scenes. The advanced user interface is using dynamic VRML objects and Java-based scripting. Web pages are represented by thumbnail images (icon instance variable of location object) with interconnections as hypertext references. Persons are shown as icons or names, which are grouped around the Web pages. Text based (chat) communication between visible persons is presented as a text track, which flows from the front to the back of the 3d-scene. Icons of psersons are click-able to initiate communication as shown in the basic user interface.

more...